Recuperación y acceso a la información

GRADO EN INGENIERÍA INFORMÁTICA

C:\Users\David\AppData\Local\Microsoft\Windows\INetCache\Content.Word\logo_UC3M_universidad_Carlos_III_Madrid.jpg

**Práctica 2**

**Modelo vectorial**

Realizado por:

Rodrigo Borges (Grupo 80-100317579)

(100317579@alumnos.uc3m.es)

David del Rey García (Grupo 80-100315243)

(100315243@alumnos.uc3m.es)

Contenido

[Introducción 2](#_Toc477294899)

[Descripción de la arquitectura 2](#_Toc477294900)

[Diagrama de clases 2](#_Toc477294901)

[CreadorDiccionario 2](#_Toc477294902)

[Calculador 3](#_Toc477294903)

[Modelos 3](#_Toc477294904)

[Resultados obtenidos 4](#_Toc477294905)

[Consulta 1 What video game won Spike's best driving game award in 2006? 4](#_Toc477294906)

[Consulta 2 What is the default combination of Kensington cables? 4](#_Toc477294907)

[Consulta 3 Who won the first ACM Gerard Salton prize? 5](#_Toc477294908)

[Mejoras implementadas 6](#_Toc477294909)

Índice de ilustraciones

[Ilustración 1. Diagrama de clases. 2](#_Toc477294891)

[Ilustración 2. Primera consulta Q1. 4](#_Toc477294892)

[Ilustración 3. Segunda consulta Q2. 4](#_Toc477294893)

[Ilustración 4. Tercera consulta Q3 5](#_Toc477294894)

[Ilustración 5. Conjunto de consultas estáticas 5](file:///C:\Users\borge\Desktop\clase\2016-2017\Segundo%20cuatri\rai\Modelo%20vectorial\ModeloVectorial_3\100315243_100317579.docx#_Toc477294895)

# Introducción

El proyecto presentado está compuesto por algoritmos que permiten conocer la relevancia que tienen ciertos documentos frente a una consulta, en función de las palabras que la componen. Concretamente se implementarán las siguientes funciones de similitud:

* Producto escalar TF.
* Producto escalar TF IDF.
* Coseno TF.
* Coseno TF IDF.

# Descripción de la arquitectura

## Diagrama de clases

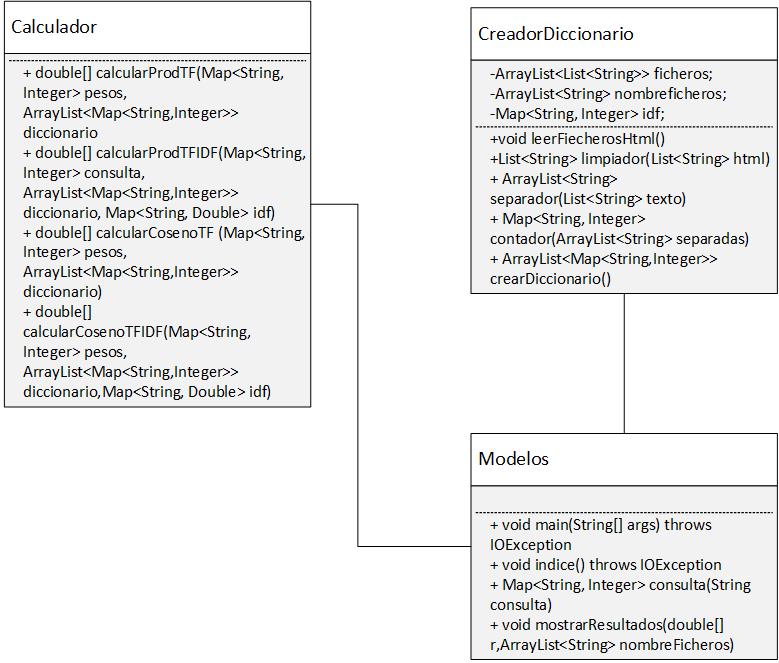


Ilustración . Diagrama de clases.

## CreadorDiccionario

Esta clase es la encargada de crear el diccionario con las palabras de los documentos.

Para poder crear el diccionario se parte de unos documentos html. Estos documentos han de estar almacenados en la ruta “*./htmls/*”, que se encuentra en la carpeta raíz del proyecto.

Una vez leídos los documentos se limpian para obtener únicamente el texto relevante de cada uno, para ello se ha utilizado la librería “*Jsoup*”; y se separan las palabras, almacenando cada palabra con su frecuencia en el documento, de manera que el diccionario se compone de páginas, y cada página representa a un documento, almacenando sus palabras con la frecuencia.

## Calculador

Esta clase es la encargada de realizar los cálculos de los índices de relevancia en función de la consulta realizada y del diccionario proporcionado.

Los índices de similitud calculados son:

* Producto escalar TF:
* Producto escalar TF-IDF:
* Producto vectorial TF:
* Producto vectorial TF-IDF:

## Modelos

Es la clase encargada de mostrar al usuario las opciones disponibles, y en función de la elección del usuario el programa realizará la acción solicitada.

Las opciones disponibles son:

* Crear/Actualizar diccionario: con esta opción el programa llama a la clase CrearDiccionario para que esta lea la ruta "./htmls/" y cree un diccionario con todos los documentos ".html" que se encuentren en ella.
* Realizar consulta: una vez creado el diccionario el usuario podrá realizar una consulta, para ello ha de seleccionar la segunda opción del índice y escribir la consulta que desee realizar. Se mostrarán los resultados por pantalla y se le volverá a dar al usuario la opción de elegir acción en el índice.
* Mostrar consultas estáticas: en la práctica se proponen 3 consultas. Esta opción realizará las consultas y las mostrará por pantalla en un formato diferente al general para poder compararlas.
* Salir: esta opción terminará la ejecución del programa.

# Resultados obtenidos

### Consulta 1 What video game won Spike's best driving game award in 2006?

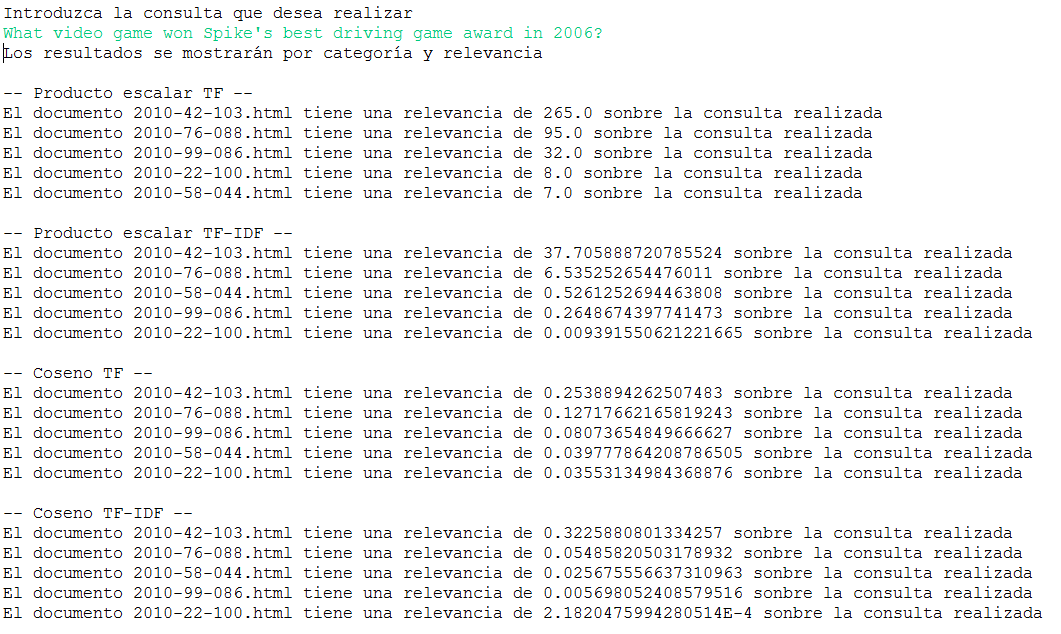


Ilustración . Primera consulta Q1.

### Consulta 2 What is the default combination of Kensington cables?

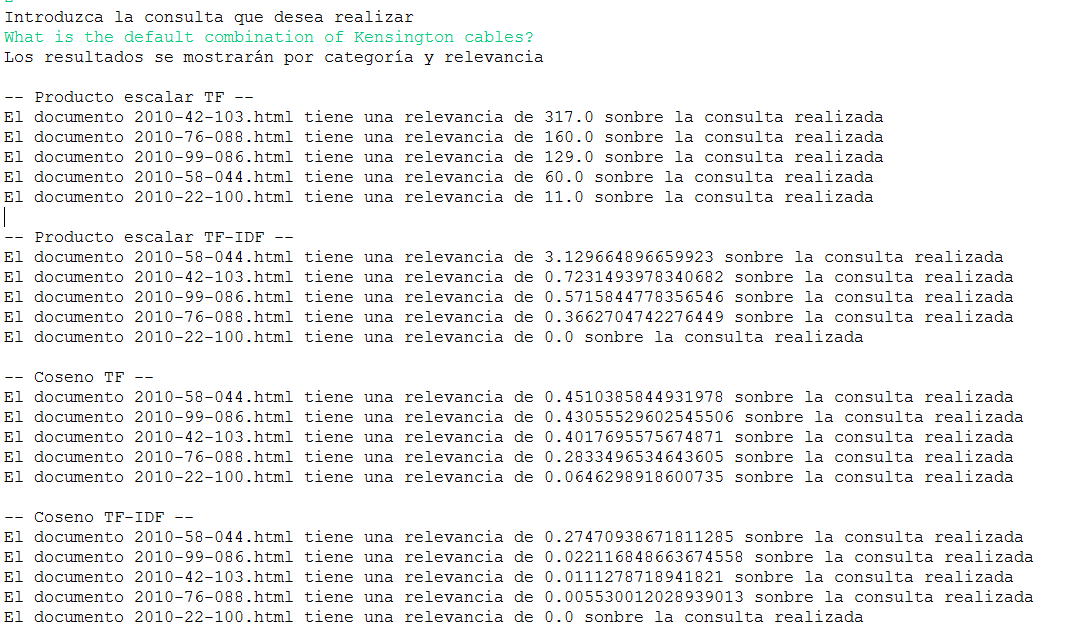


Ilustración . Segunda consulta Q2.

### Consulta 3 Who won the first ACM Gerard Salton prize?

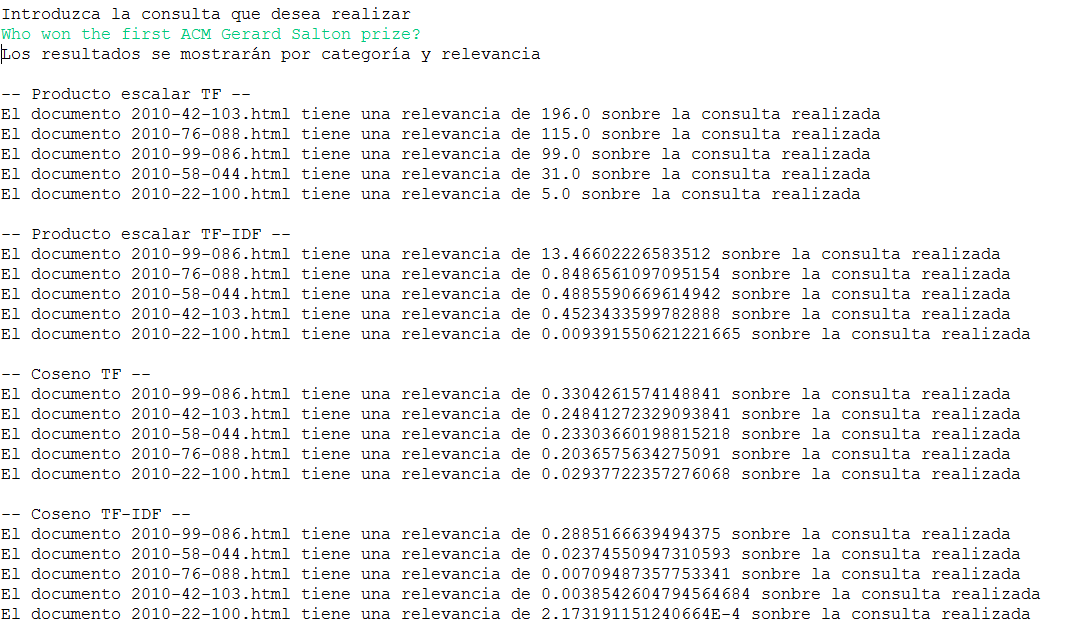


Ilustración . Tercera consulta Q3

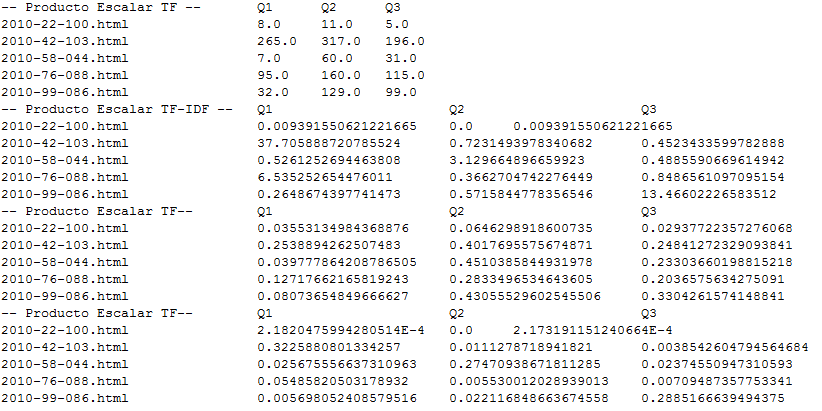
En la siguiente imagen se muestra la relevancia de las 3 consultas propuestas en los 5 documentos “*.html*”. No están ordenados por relevancia como en las consultas más generales.

Ilustración . Conjunto de consultas estáticas

Como se puede ver en los resultados, hay ficheros que parecen más relevantes frente a los demás, en algunos índices de similitud. Esto se debe a que poseen mayor cantidad de texto, por lo que se dan más coincidencias de palabras de las consultas en los textos, la mayoría palabras carentes de valor. Por ello una de las posibles mejoras sería la eliminación de palabras carentes de valor para que las consultas fuesen más precisas. En cambio, al no haber realizado dicha mejora aún, hay ciertos ficheros que no son relevantes para las consultas realizadas pero aparecen como relevantes al tener un alto número de coincidencias.

# Mejoras implementadas

Normalización de términos: tanto para los documentos como para las consultas hemos eliminado los caracteres especiales (solo mantenemos caracteres alfanuméricos) y transformado a minúsculas todos los términos. Esta es la primera aproximación a la normalización de términos, en la siguiente versión se reducirá cada termino a su raíz.

Limpieza de etiquetas: para llevar a cabo la eliminación de las etiquetas hemos utilizado la librería Jsoup de java. Durante el desarrollo tuvimos problemas con la eliminación del símbolo ‘>’ al final de las etiquetas, si no existía un espacio entre texto etiqueta y texto Jsoup nos juntaba las palabras. Para solucionar este problema añadimos un espacio cada vez que encontramos el símbolo ‘>’ y de esta manera evitamos que nos concatenara palabras que debían estar separadas.

Separación de cálculo de pesos de la función de similitud: los valores de TF e IDF se calculan durante la creación del diccionario y las funciones de similitud se calculan utilizando la clase calculador, cada función de similitud tiene su propio método. En la siguiente versión se re-factorizara para restar responsabilidad la clase calculador.

Filtrado de palabras vacías: no se ha realizado ningún filtrado sobre estas palabras. En la siguiente versión se incluirá esta funcionalidad.